SIEMENS

Application Note 39

USB Interface Description

Siemens Cellular Engine

Version: 02
DocId: HC15_USB_Interface_v02

Products: HC15

AN 39:	**USB Interface Description**
Version:	**02**
Date:	**2007-04-05**
DocId:	**HC15_USB_Interface_v02**
Status	**Confidential / Preliminary**
Supported Products:	**HC15**

General Notes

Product is deemed accepted by Recipient and is provided without interface to Recipient´s products. The Product constitutes pre-release version and code and may be changed substantially before commercial release. The Product is provided on an “as is” basis only and may contain deficiencies or inadequacies. The Product is provided without warranty of any kind, express or implied. To the maximum extent permitted by applicable law, Siemens further disclaims all warranties, including without limitation any implied warranties of merchantability, fitness for a particular purpose and noninfringement of third-party rights. The entire risk arising out of the use or performance of the Product and documentation remains with Recipient. This Product is not intended for use in life support appliances, devices or systems where a malfunction of the product can reasonably be expected to result in personal injury. Applications incorporating the described product must be designed to be in accordance with the technical specifications provided in these guidelines. Failure to comply with any of the required procedures can result in malfunctions or serious discrepancies in results. Furthermore, all safety instructions regarding the use of mobile technical systems, including GSM products, which also apply to cellular phones must be followed. Siemens AG customers using or selling this product for use in any applications do so at their own risk and agree to fully indemnify Siemens for any damages resulting from illegal use or resale. To the maximum extent permitted by applicable law, in no event shall Siemens or its suppliers be liable for any consequential, incidental, direct, indirect, punitive or other damages whatsoever (including, without limitation, damages for loss of business profits, business interruption, loss of business information or data, or other pecuniary loss) arising out the use of or inability to use the Product, even if Siemens has been advised of the possibility of such damages. Subject to change without notice at any time.

Copyright

Transmittal, reproduction, dissemination and/or editing of this document as well as utilization of its contents and communication thereof to others without express authorization are prohibited. Offenders will be held liable for payment of damages. All rights created by patent grant or registration of a utility model or design patent are reserved.

Copyright © Siemens AG 2007

Contents

0 Document History 5

1 Introduction 6
1.1 Related Documents 6
1.2 Abbreviations 6

2 Overview 7

3 USB Interface Functionality 8
3.1 Start Up 9
3.2 Composite Device Interfaces for "MdmNet" Enumeration 10
3.2.1 Interface 0 (VCOM Port) 10
3.2.2 Interface 1 (NET Port) 10
3.2.3 Interface 2 (MODEM Port) 10
3.3 Composite Device Interfaces for "Mdm" Enumeration 11
3.3.1 Interfaces 0 and 1 (CDC ACM MODEM Port) 11
3.3.1.1 Interface 0 (CDC ACM – Control) 11
3.3.1.2 Interface 1 (CDC ACM – Data) 11
3.3.2 Interface 2 (VCOM1 Port) 11
3.3.3 USB Interface 3 (VCOM2 Port) 11
3.3.4 Usage of Non-Windows USB Host Drivers 12
3.4 Transition from Composite to Mass Storage Device 13
3.4.1 Transition Timer 13

4 USB Interface Descriptors 14
4.1 Composite Device Enumeration “MdmNet“ Variant 14
4.1.1 Device Descriptor 14
4.1.2 Configuration Descriptor 15
4.1.2.1 Configuration 15
4.1.2.2 Interface 0 (VCOM Port) 15
4.1.2.3 Interface 1 (NET Port) 17
4.1.2.4 Interface 2 (MODEM Port) 19
4.1.3 String Descriptor 21
4.1.3.1 String 0 - Language IDs 21
4.1.3.2 String 1 - iManufacturer 21
4.1.3.3 String 2 - iProduct 21
4.1.3.4 String 3 - iInterface 21
4.1.3.5 String 4 - iInterface 22
4.1.3.6 String 5 - iInterface 22

4.2 Composite Device Enumeration for “Mdm“ Variant 23
4.2.1 Device Descriptor 23
4.2.2 Configuration Descriptor 24
4.2.2.1 Configuration 24
4.2.2.2 Interface 0 (MODEM Port: CDC ACM Control) 24
4.2.2.3 Interface 1 (MODEM port: CDC ACM Data) 25
4.2.2.4 Interface 2 (VCOM1 Port) 26
4.2.2.5 Interface 3 (VCOM2 Port) 27
4.2.3 String Descriptor 28

0 Document History

New document: “AN39: USB Interface Description“, v02
Preceding document: “AN39: USB Interface Description“, v01

Chapter	What is new
Throughout document	Added alternative composite device enumeration.

New document: “AN39: USB Interface Description“, v01

Chapter	What is new
---	Initial document setup.

1 Introduction

This document describes HC15's USB interface. Its purpose is to support application engineers in developing a USB host driver for the module's USB interface.

1.1 Related Documents

Documents supplied with your Siemens cellular engine
[1] AT Command Set supplied with your Siemens cellular engine
[2] Hardware Interface Description supplied with your Siemens cellular engine
[3] Release Notes supplied with your Siemens cellular engine
[4] Getting Started with HC15

Other related documents
[5] Universal Serial Bus Specification Revision 2.0 April 27, 2000,
[6] USB Class Definitions for Communication Devices Version 1.1 January 19, 1999

1.2 Abbreviations

Abbreviation	Description
CDC ACM	Communication Device Class Abstract Control Model
GPRS	General Packet Radio Service
HSDPA	High-Speed Downlink Packet Access
ME	Mobile Equipment
MO	Mobile Originated
MT	Mobile Terminated
SE0	Single ended 0
SOF packet	Start of Frame packet
URC	Unsolicited Result Code
USB	Universal Serial Bus
VCOM port	Virtual Communication port

2 Overview

HC15's USB interface by default enumerates, i.e., registers with the USB host, as a *USB composite device* supporting three separate interfaces. Alternatively, the module's USB interface can also enumerate either as a composite device with four interfaces or as a *USB mass storage device*. All of these USB interface device classes are exclusive. It is non-volatile configurable at run-time which device class is activated as soon as the module is switched on (see Chapter 3).

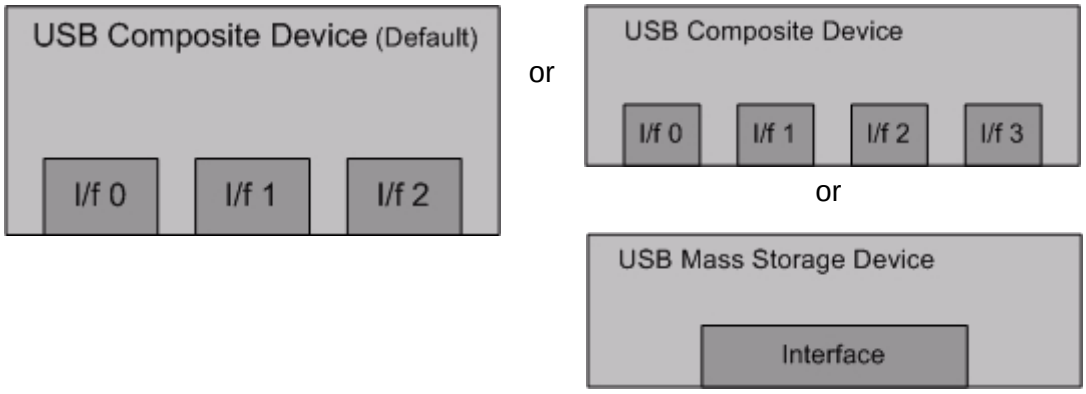

Figure 1: USB interface device classes

Under Windows XP the default USB composite device ("MdmNet" enumeration) implements a virtual COM port (VCOM) and a modem port (MODEM) that are based on the Communication Device Class Abstract Control Model (CDC ACM) specified in [6] as well as a wireless network adapter (NET) - see Section 3.2. Each interface is controlled by a separate device driver running on the Windows XP host. These drivers are stored on the module and need to be installed before the device can be used (see [4]).

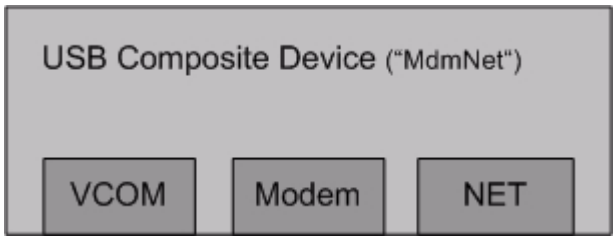

Figure 2: Default USB Host Driver Structure for Windows XP

The alternative USB composite communication device ("Mdm" enumeration) implements four USB interfaces that are mapped to three virtual ports on the host: a CDC ACM compliant modem port (MODEM), a virtual COM port (VCOM) and a second virtual COM port that is currently reserved for future use - see Section 3.3).

The mass storage device is supported solely for the purpose of facilitating the USB host driver installation without having to resort to an external storage media like CD-ROM. As long as there is no USB composite device host driver installed or active, the USB interface will automatically re-enumerate as mass storage device after a configurable period of time. This transition mechanism is described in more detail in Section 3.4. After setting up the USB composite device host driver software from the mass storage device, the USB interface automatically becomes an "MdmNet" enumerated USB composite device again. This is realized by the setup software through issueing a mass storage device class termination request to close the mass storage device and thus triggering re-enumeration as USB composite device.

3 USB Interface Functionality

HC15 supports a USB 2.0 Full Speed (12Mbit/s) device interface (see [2]) with the following default configuration settings:

Table 1: Default configuration settings for USB interface

Parameter	Description	Default Setting (Factory)
startup	USB interface configuration at startup. Default setting: "MdmNet" enumerated USB composite device.	"MdmNet"
maxpow	Maximum power consumption of the USB device from the USB bus (in mA).	10
powsrc	Power source for USB device is either bus or self powered.	"SELF"
mnto	Timeout for the communication device class request (in ms).	10000
mswprot	Write protection for mass storage device is either enabled or disabled.	"Disabled"
msoe	USB interface configuration on ejecting the module enumerated as mass storage device. Default setting: "MdmNet" enumerated USB composite device.	"MdmNet"

The USB interface can be configured non-volatile by means of AT command (see [1], AT^SUSB).

3.1 Start Up

By default, HC15's USB interface is invoked as an "MdmNet" enumerated USB composite device after startup. For an overview of HC15's USB interface descriptors see Chapter 4.

After enumeration as USB composite device and as soon as the module's USB driver has reached the state "configured" (see [5] for USB state description), the interface endpoints (i.e., all endpoints except 0) are stalled. The interface endpoints can only be addressed after a CLEAR_FEATURE (ENDPOINT_HALT) according to [5] has been issued by the host for this endpoint. Endpoints with both IN and OUT pipe should be enabled by issuing the CLEAR_FEATURE (ENDPOINT_HALT) request for IN.

In case HC15 is configured to start up as a mass storage device only, the following steps are required in order to re-enumerate as "MdmNet" or "Mdm" variant:

- Eject the mass storage device. Under Windows XP this is done by right-clicking on the assigned *Removable Disk* drive and selecting *Eject*. The module's USB interface then re-enumerates as composite device and the host driver software is able to access the interface devices.
- If need be, re-configure the mass storage startup behavior by means of AT command and restart the module as USB composite device (see [1], AT^SUSB).

For HC15's USB interface to be invoked as an "Mdm" enumerated USB composite device, the following steps are recommended as long as it is not possible to communicate with HC15 in any other way:

- Install the USB composite device host driver software for the "MdmNet" variant on a Windows XP system to be able to initially access the module (see [4]). Please note that the supplied HC15 USB driver package has been designed for Windows XP only.
- Re-configure the module by means of AT command to start up with an "Mdm" enumeration (see [1], AT^SUSB).
- Now, the module can be connected to and restarted on other (non-Windows) host systems with USB host drivers compatible with the Communication Device Class Abstract Control Model specification (CDC ACM, see [6]). For a description of how to employ the standard Linux USB drivers see Section 3.3.4.

Note: Please do not specify "0" for the AT^SUSB parameter <mnto> and do not change the default value "MdmNet" for the parameter <msoe>, unless you are sure to have installed the correct USB host driver software for your USB host system. This allows a fall back to the "MdmNet" enumeration in case the driver software is not functional: The USB mass storage device is invoked after the configured timeout (see Section 3.4) and the "MdmNet" enumeration which Siemens provides Windows XP host driver software for can be activated by means of ejecting the mass storage device.

3.2 Composite Device Interfaces for "MdmNet" Enumeration

3.2.1 Interface 0 (VCOM Port)

Interface 0 implements two endpoints:
- Bulk-only data IN
- Bulk-only data OUT endpoint

Using the supplied host driver software for Windows XP this interface is implemented as a virtual USB COM port and serves as so-called application port. The application port is intended for controlling the module via AT commands or for receiving URCs. It can be used also for sending, receiving, writing and reading short messages. The application port is not intended as HSDPA or GPRS data interface.

3.2.2 Interface 1 (NET Port)

Interface 1 implements three endpoints:
- a bulk-only data IN,
- a bulk-only data OUT
- an interrupt IN notification endpoint

Using the supplied host driver software for Windows XP this interface serves as a wireless network adapter intended for HSDPA and GPRS data connections. For details on how to use this interface see [4].

3.2.3 Interface 2 (MODEM Port)

Interface 2 implements three endpoints:
- Bulk-only data IN,
- Bulk-only data OUT
- Interrupt IN notification endpoint

Using the supplied host driver software for Windows XP this interface is implemented as a HSDPA USB modem and serves as so-called modem port. The modem port is particularly intended for HSDPA and GPRS data connections. It can be used to set up dial-up network connections.

Notes:
- The AT command parameter AT^SCFG="URC/DstIfc" can be configured to redirect output URCs from the default and recommended Interface 0 (VCOM application port) to the modem port (see [1], AT^SCFG). If the modem interface is used for redirected URCs and blocked by an active data connection, the data connection should be suspended from time to time to check for pending URCs.
- The interface's notification endpoint is also employed for notifications from the interface 0 (VCOM). A USB host driver should therefore evaluate the interface number of the notification interface (see [6], Section 6.3).

3.3 Composite Device Interfaces for "Mdm" Enumeration

3.3.1 Interfaces 0 and 1 (CDC ACM MODEM Port)

This port is mainly intended for dialup data connections. It implements a standard legacy modem protocol. The port consists of two USB interfaces. One interface serves as CDC ACM control interface, the other one as CDC ACM data interface.

3.3.1.1 Interface 0 (CDC ACM – Control)

Interface 0 has only one endpoint:
- Interrupt IN notification endpoint

The MODEM notification endpoint is also used for notifications from VCOM1 port (see Section 3.3.2). Therefore, an USB host driver must evaluate the notification interface number.

3.3.1.2 Interface 1 (CDC ACM – Data)

Interface 1 has two endpoints:
- Bulk-only data IN
- Bulk-only data OUT

3.3.2 Interface 2 (VCOM1 Port)

Interface 2 has two endpoints:
- Bulk-only data IN
- Bulk-only data OUT

The interface serves as so-called application port and supports AT commands as specified in [1] or [2], but no data connections.

3.3.3 USB Interface 3 (VCOM2 Port)

Interface 3 has two endpoints:
- Bulk-only data IN
- Bulk-only data OUT

Interface 3 is only enumerated. It is reserved for possible future use.

3.3.4 Usage of Non-Windows USB Host Drivers

As a rule the "Mdm" enumeration is compatible with generic CDC ACM based USB host drivers.

However, to employ generic USB host drivers, specific issues may have to be taken into account depending on the operating system:

- On a Linux system - i.e., Linux 2.6 kernel with loadable module support and USB host drivers "cdc_acm" as well as "usbserial" compiled as modules - the "cdc_acm" driver should be loaded prior to "usbserial" to ensure a correct enumeration. The "cdc_acm" driver attaches the CDC ACM MODEM port to /dev/ttyACMx. Also, the "usbserial" driver should be loaded with the parameters:
  modprobe usbserial vendor=0x0681 product=0x0046

  This attaches /dev/ttyUSBx and /dev/ttyUSB(x+1). ttyUSBx is the VCOM1 application port, the other VCOM port is currently unused.

  To load the USB drivers automatically, a script realizing the above issues could be added to the other startup scripts - usually under etc/init.d - and the runlevels updated accordingly.

3.4 Transition from Composite to Mass Storage Device

As a rule the module's USB interface is invoked as an "MdmNet" or "Mdm" enumerated USB composite device (see Table 1; parameter "startup"). However, if no USB composite host driver is installed or active, the host will not send any USB communication device class request after enumeration (see Section 3.4.1). Instead, it only sends SOF packets (=Start-Of-Frame packets; sent every 1ms for a USB full speed interface).

In this case, the module will re-enumerate as a USB mass storage device after a specified time period (see Table 1; parameter "mnto"). The mass storage device holds the Windows XP host driver software for the default USB composite device. The following section describes the timing mechanism that controlls the transition from composite to mass storage device. This mechanism is enabled only if the module is configured to initially enumerate as a composite communication device ("MdmNet" or "Mdm"). See note in Section 3.1 for the steps required to terminate the mass storage device.

3.4.1 Transition Timer

The transition from composite to mass storage device is controlled by a timer configured for the module (see Table 1; parameter "mnto").

The USB host starts the timer with its initial GET_DESCRIPTOR (DEVICE) request (according to [5], Section 9.4.3) to which the module replies with the appropriate device descriptor (see Chapter 4). Any further GET_DESCRIPTOR (DEVICE) request received within the configured time period, i.e., while the timer is running, will restart the timer. Any GET_DESCRIPTOR (DEVICE) request received after the timer has expired does not restart the timer. The timer is only started once after power up.

Note: The timing mechanism operates with low priority in the firmware task context. The configured time period does therefore only represent the minimum period of time before the timer expires. The exact period cannot be guaranteed. If an application is busy with higher priority tasks, the timeout may be delayed for some 10ms. Considerably longer periods are also possible.

If the configured time period (i.e., the USB communication device class request timeout) expires without any of the USB requests listed below being received by the module, a USB SE0 condition (see [5], Section 7.1.7.1) is driven on the bus by the module for t > 10ms (typical 200ms) to trigger re-enumeration by the USB host, i.e., a restart as mass storage device:

- CLEAR_FEATURE (ENDPOINT)
  This is actually no communication class request but a standard request according to [5], Section 9.4.1. Because this request is mandatory to get the endpoints associated with communication device interfaces into operation, it is also an indicator for a successfully installed USB modem host driver.
- SET_LINE_CODING
  Communication class request according to [6], Section 6.2.12
- GET_LINE_CODING
  Communication class request according to [6], Section 6.2.13
- SET_CONTROL_LINE_STATE
  Communication class request according to [6], Section 6.2.14

Only if any of the mentioned USB requests are received within the configured time period will the module remain enumerated as "MdmNet" or "Mdm" USB composite device.

4 USB Interface Descriptors

This chapter lists the USB interface descriptors for the implemented USB composite device enumerations. The default “MdmNet“ enumeration implements three device interfaces - VCOM port, NET port, MODEM port (see Section 4.1), whereas the “Mdm“ enumeration implements four interfaces that are mapped to three virtual ports on the host: a CDC ACM compliant MODEM port as well as two VCOM ports where the second VCOM port is currently reserved for future use (see Section 4.2).

4.1 Composite Device Enumeration “MdmNet“ Variant

4.1.1 Device Descriptor

Table 2: MDMNET\Device Descriptor HC15

Name	Value	Dec	Hex
bLength	Valid	18	0x12
bDescriptorType	DEVICE	1	0x01
bcdUSB	1.1	272	0x0110
bDeviceClass	Communication	2	0x02
bDeviceSubClass	None	0	0x00
bDeviceProtocol	None	0	0x00
bMaxPacketSize0	64	64	0x40
idVendor	Siemens AG	1665	0x0681
idProduct	62	62	0x003E
bcdDevice	Depending on USB firmware version		
iManufacturer	1	1	0x01
iProduct	2	2	0x02
iSerialNumber	No serial number	0	0x00
bNumConfigurations	1	1	0x01

4.1.2 Configuration Descriptor

4.1.2.1 Configuration

Table 3: MDMNET\Configuration Descriptor\Configuration

Name	Value	Dec	Hex
bLength	Valid	9	0x09
bDescriptorType	CONFIGURATION	2	0x02
wTotalLength	173 bytes	173	0x00AD
bNumInterface	3	3	0x03
bConfigurationValue	1	1	0x01
iConfiguration	0	0	0x00
bmAttributes.SelfPowered	Depending on configuration [1] (see [1]; AT^SUSB)		
bmAttributes.RemoteWakeup	Supported		
bMaxPower	Depending on configuration [1] (see [1]; AT^SUSB)	0..255	0x00..0xFA

1. If the firmware download fails - e.g., due to a power supply disconnect during download - the module’s boot loader enumerates as bus powered (with bMaxPower=0xFA (500mA), irrespective of the current descriptor setting.

4.1.2.2 Interface 0 (VCOM Port)

Table 4: MDMNET\Configuration Descriptor\Interface0

Name	Value	Dec	Hex
bLength	Valid	9	0x09
bDescriptorType	INTERFACE	4	0x04
bInterfaceNumber	0	0	0x00
bAlternateSetting	0	0	0x00
bNumEndpoints	2	2	0x02
bInterfaceClass	Vendor-specific	255	0xFF
bInterfaceSubClass	Vendor-specific	255	0xFF
bInterfaceProtocol	Vendor-specific	255	0xFF
iInterface	5	5	0x05

Vendor-specific Descriptor 1

Table 5: MDMNET\Configuration Descriptor\Interface0\Vendor-specific Descriptor 1

Name	Value	Dec	Hex
bFunctionLength	Valid	21	0x15
bDescriptorType	CS_INTERFACE	36	0x24
Vendor specific elements up to full *bfunctionLength*, i.e., 19 more elements.	Vendor specific	. . .	. . .

Vendor-specific Descriptor 2

Table 6: MDMNET\Configuration Descriptor\Interface0\Vendor-specific Descriptor 2

Name	Value	Dec	Hex
bFunctionLength	Valid	6	0x06
bDescriptorType	CS_INTERFACE	36	0x24
Vendor specific elements up to full *bfunctionLength*, i.e., 4 more elements.	Vendor specific	. . .	. . .

Endpoint Descriptor 1

Table 7: MDMNET\Configuration Descriptor\Interface0\Endpoint Descriptor

Name	Value	Dec	Hex
bFunctionLength	Valid	7	0x07
bDescriptorType	ENDPOINT	5	0x05
bEndpointAddress	4 IN [1]	132	0x84
bmAttributes.TransferType	Bulk	2	0x02
wMaxPacketSize	64	64	0x0040
bInterval	Ignored	0	0x00

1. The specified endpoint address is an example and can be changed without notice.

Endpoint Descriptor 2

Table 8: MDMNET\Configuration Descriptor\Interface0\Endpoint Descriptor

Name	Value	Dec	Hex
bFunctionLength	Valid	7	0x07
bDescriptorType	ENDPOINT	5	0x05
bEndpointAddress	4 OUT [1*)]	4	0x04
bmAttributes.TransferType	Bulk	2	0x02
wMaxPacketSize	64	16	0x0010
bInterval	Ignored	0	0x00

1. The specified endpoint address is an example and can be changed without notice.

4.1.2.3 Interface 1 (NET Port)

Table 9: MDMNET\Configuration Descriptor\Interface1

Name	Value	Dec	Hex
bLength	Valid	9	0x09
bDescriptorType	INTERFACE	4	0x04
bInterfaceNumber	1	1	0x01
bAlternateSetting	0	0	0x00
bNumEndpoints	3	3	0x03
bInterfaceClass	Vendor-specific	255	0xFF
bInterfaceSubClass	Vendor-specific	0	0xFF
bInterfaceProtocol	Vendor-specific	0	0xFF
iInterface	4	4	0x04

Vendor-specific Descriptor 1

Table 10: MDMNET\Configuration Descriptor\Interface1\Vendor-specific Descriptor 1

Name	Value	Dec	Hex
bFunctionLength	Valid	21	0x15
bDescriptorType	CS_INTERFACE	36	0x24
Vendor specific elements up to full *bfunctionLength*, i.e., 19 more elements.	Vendor specific	. . .	. . .

Vendor-specific Descriptor 2

Table 11: MDMNET\Configuration Descriptor\Interface1\Vendor-specific Descriptor 2

Name	Value	Dec	Hex
bFunctionLength	Valid	6	0x06
bDescriptorType	CS_INTERFACE	36	0x24
Vendor specific elements up to full *bfunctionLength*, i.e., 4 more elements.	Vendor specific	. . .	. . .

Endpoint Descriptor 1

Table 12: MDMNET\Configuration Descriptor\Interface1\Endpoint Descriptor

Name	Value	Dec	Hex
bFunctionLength	Valid	7	0x07
bDescriptorType	ENDPOINT	5	0x05
bEndpointAddress	5 IN [1]	133	0x85
bmAttributes.TransferType	Interrupt	3	0x03
wMaxPacketSize	16	16	0x0010
bInterval	128 frames	128	0x80

1. The specified endpoint address is an example and can be changed without notice.

Endpoint Descriptor 2

Table 13: MDMNET\Configuration Descriptor\Interface1\Endpoint Descriptor

Name	Value	Dec	Hex
bFunctionLength	Valid	7	0x07
bDescriptorType	ENDPOINT	5	0x05
bEndpointAddress	3 IN [1]	131	0x83
bmAttributes.TransferType	Bulk	2	0x02
wMaxPacketSize	64	64	0x0040
bInterval	Ignored	0	0x00

1. The specified endpoint address is an example and can be changed without notice.

Endpoint Descriptor 3

Table 14: MDMNET\Configuration Descriptor\Interface1\Endpoint Descriptor

Name	Value	Dec	Hex
bFunctionLength	Valid	7	0x07
bDescriptorType	ENDPOINT	5	0x05
bEndpointAddress	3 OUT [1]	3	0x03
bmAttributes.TransferType	Bulk	2	0x02
wMaxPacketSize	64	64	0x0040
bInterval	Ignored	0	0x00

1. The specified endpoint address is an example and can be changed without notice.

4.1.2.4 Interface 2 (MODEM Port)

Table 15: MDMNET\Configuration Descriptor\Interface2

Name	Value	Dec	Hex
bLength	Valid	9	0x09
bDescriptorType	INTERFACE	4	0x04
bInterfaceNumber	2	2	0x02
bAlternateSetting	0	0	0x00
bNumEndpoints	3	3	0x03
bInterfaceClass	Vendor-specific	255	0xFF
bInterfaceSubClass	Vendor-specific	255	0xFF
bInterfaceProtocol	Vendor-specific	255	0xFF
iInterface	3	3	0x03

Vendor-specific Descriptor 1

Table 16: MDMNET\Configuration Descriptor\Interface2\Vendor-specific Descriptor 1

Name	Value	Dec	Hex
bFunctionLength	Valid	21	0x15
bDescriptorType	CS_INTERFACE	36	0x24
Vendor specific elements up to full *bfunctionLength*, i.e., 19 more elements.	Vendor specific	. . .	. . .

Vendor-specific Descriptor 2

Table 17: MDMNET\Configuration Descriptor\Interface2\Vendor-specific Descriptor 2

Name	Value	Dec	Hex
bFunctionLength	Valid	6	0x06
bDescriptorType	CS_INTERFACE	36	0x24
Vendor specific elements up to full *bfunctionLength*, i.e., 4 more elements.	Vendor specific	. . .	. . .

Endpoint Descriptor 1

Table 18: MDMNET\Configuration Descriptor\Interface2\Endpoint Descriptor

Name	Value	Dec	Hex
bFunctionLength	Valid	7	0x07
bDescriptorType	ENDPOINT	5	0x05
bEndpointAddress	1 IN [1]	129	0x81
bmAttributes.TransferType	Interrupt	3	0x03
wMaxPacketSize	64	64	0x0040
bInterval	128 frames	128	0x80

1. The specified endpoint address is an example and can be changed without notice.

Endpoint Descriptor 2

Table 19: MDMNET\Configuration Descriptor\Interface2\Endpoint Descriptor

Name	Value	Dec	Hex
bFunctionLength	Valid	7	0x07
bDescriptorType	ENDPOINT	5	0x05
bEndpointAddress	2 IN [1]	130	0x82
bmAttributes.TransferType	Bulk	2	0x02
wMaxPacketSize	64	64	0x0040
bInterval	Ignored	0	0x00

1. The specified endpoint address is an example and can be changed without notice.

Endpoint Descriptor 3

Table 20: MDMNET\Configuration Descriptor\Interface2\Endpoint Descriptor

Name	Value	Dec	Hex
bFunctionLength	Valid	7	0x07
bDescriptorType	ENDPOINT	5	0x05
bEndpointAddress	2 OUT [1]	2	0x02
bmAttributes.TransferType	Bulk	2	0x02
wMaxPacketSize	64	64	0x0040
bInterval	Ignored	0	0x00

1. The specified endpoint address is an example and can be changed without notice.

4.1.3 String Descriptor

The string descriptor values specified below are examples and can be changed without notice.

4.1.3.1 String 0 - Language IDs

Table 21: MDMNET\String Descriptor\Language IDs

Name	Value	Dec	Hex
bFunctionLength	Valid	4	0x04
bDescriptorType	STRING	3	0x03
wLANGID[0]	English (US)	1033	0x0409

4.1.3.2 String 1 - iManufacturer

Table 22: MDMNET\String Descriptor\iManufacturer

Name	Value	Dec	Hex
bFunctionLength	Valid	20	0x14
bDescriptorType	STRING	3	0x03
	“Siemens AG”		

4.1.3.3 String 2 - iProduct

Table 23: MDMNET\String Descriptor\iProduct

Name	Value	Dec	Hex
bFunctionLength	Valid	26	0x1A
bDescriptorType	STRING	3	0x03
	“Siemens HC15”		

4.1.3.4 String 3 - iInterface

Table 24: MDMNET\String Descriptor\iInterface

Name	Value	Dec	Hex
bFunctionLength	Valid	22	0x16
bDescriptorType	STRING	3	0x03
	“HC15 MODEM”		

4.1.3.5 String 4 - iInterface

Table 25: MDMNET\String Descriptor\iInterface

Name	Value	Dec	Hex
bFunctionLength	Valid	18	0x12
bDescriptorType	STRING	3	0x03
	“HC15 NET”		

4.1.3.6 String 5 - iInterface

Table 26: MDMNET\String Descriptor\iInterface

Name	Value	Dec	Hex
bFunctionLength	Valid	20	0x14
bDescriptorType	STRING	3	0x03
	“HC15 VCOM”		

4.2 Composite Device Enumeration for “Mdm“ Variant

4.2.1 Device Descriptor

Table 27: MDM\Device Descriptor

Name	Value	Dec	Hex
bLength	Valid	18	0x12
bDescriptorType	DEVICE	1	0x01
bcdUSB	1.1	272	0x0110
bDeviceClass	Communication	2	0x02
bDeviceSubClass	None	0	0x00
bDeviceProtocol	None	0	0x00
bMaxPacketSize0	64	64	0x40
idVendor	Siemens AG	1665	0x0681
idProduct	70	70	0x0046
bcdDevice	Dependent on USB firmware version		
iManufacturer	1	1	0x01
iProduct	2	2	0x02
iSerialNumber	0	0	0x00
bNumConfigurations	1	1	0x01

4.2.2 Configuration Descriptor

4.2.2.1 Configuration

Table 28: MDM\Configuration Descriptor\Configuration

Name	Value	Dec	Hex
bLength	Valid	9	0x09
bDescriptorType	CONFIGURATION	2	0x02
wTotalLength	113 bytes	113	0x0071
bNumInterface	4	4	0x04
bConfigurationValue	1	1	0x01
iConfiguration	0	0	0x00
bmAttributes.SelfPowered	Dependent on configuration [1] (see [1]; AT^SUSB)		
bmAttributes.RemoteWakeup	Supported		
bMaxPower	Dependent on configuration [1] (see [1]; AT^SUSB)	0..255	0x00..0xFA

1. If the firmware download fails - e.g., due to a power supply disconnect during download - the module’s boot loader enumerates as bus powered (with bMaxPower=0xFA (500mA), irrespective of the current descriptor setting.

4.2.2.2 Interface 0 (MODEM Port: CDC ACM Control)

Table 29: MDM\Configuration Descriptor\Interface0

Name	Value	Dec	Hex
bLength	Valid	9	0x09
bDescriptorType	INTERFACE	4	0x04
bInterfaceNumber	0	0	0x00
bAlternateSetting	0	0	0x00
bNumEndpoints	1	1	0x01
bInterfaceClass	Communication Control	2	0x02
bInterfaceSubClass	Abstract Line Control Model	2	0x02
bInterfaceProtocol	Common AT commands	1	0x01
iInterface	3	3	0x03

Endpoint Descriptor

Table 30: MDM\Configuration Descriptor\Interface0\Endpoint Descriptor

Name	Value	Dec	Hex
bFunctionLength	Valid	7	0x07
bDescriptorType	ENDPOINT	5	0x05
bEndpointAddress	1 IN [1]	129	0x81
bmAttributes.TransferType	Interrupt	3	0x03
wMaxPacketSize	16	16	0x0010
bInterval	32 frames (32ms) [2]	32	0x20

1. Given endpoint address is an example and can be changed without notice.
2. Given interval is an example and can be changed without notice.

4.2.2.3 Interface 1 (MODEM port: CDC ACM Data)

Table 31: MDM\Configuration Descriptor\Interface1

Name	Value	Dec	Hex
bLength	Valid	9	0x09
bDescriptorType	INTERFACE	4	0x04
bInterfaceNumber	1	1	0x01
bAlternateSetting	0	0	0x00
bNumEndpoints	2	2	0x02
bInterfaceClass	Communication Data	10	0x0A
bInterfaceSubClass		0	0x00
bInterfaceProtocol	None	0	0x00
iInterface	3	3	0x03

Endpoint Descriptor

Table 32: MDM\Configuration Descriptor\Interface1\Endpoint Descriptor

Name	Value	Dec	Hex
bFunctionLength	Valid	7	0x07
bDescriptorType	ENDPOINT	5	0x05
bEndpointAddress	2 IN [1]	130	0x82
bmAttributes.TransferType	Bulk	2	0x02
wMaxPacketSize	64	64	0x0040
bInterval		0	0x00

1. Given endpoint address is an example and can be changed without notice.

Endpoint Descriptor

Table 33: MDM\Configuration Descriptor\Interface1\Endpoint Descriptor

Name	Value	Dec	Hex
bFunctionLength	Valid	7	0x07
bDescriptorType	ENDPOINT	5	0x05
bEndpointAddress	2 OUT [1*)]	2	0x02
bmAttributes.TransferType	Bulk	2	0x02
wMaxPacketSize	64	64	0x0040
bInterval		0	0x00

1. Given endpoint address is an example and can be changed without notice.

4.2.2.4 Interface 2 (VCOM1 Port)

Table 34: MDM\Configuration Descriptor\Interface2

Name	Value	Dec	Hex
bLength	Valid	9	0x09
bDescriptorType	INTERFACE	4	0x04
bInterfaceNumber	2	2	0x02
bAlternateSetting	0	0	0x00
bNumEndpoints	2	2	0x02
bInterfaceClass	Vendor-specific	255	0xFF
bInterfaceSubClass	Vendor-specific	255	0xFF
bInterfaceProtocol	Vendor-specific	255	0xFF
iInterface	3	3	0x03

Endpoint Descriptor

Table 35: MDM\Configuration Descriptor\Interface2\Endpoint Descriptor

Name	Value	Dec	Hex
bFunctionLength	Valid	7	0x07
bDescriptorType	ENDPOINT	5	0x05
bEndpointAddress	4 IN [1]	132	0x84
bmAttributes.TransferType	Bulk	2	0x02
wMaxPacketSize	64	64	0x0040
bInterval		0	0x00

1. Given endpoint address is an example and can be changed without notice.

Endpoint Descriptor

Table 36: MDM\Configuration Descriptor\Interface2\Endpoint Descriptor

Name	Value	Dec	Hex
bFunctionLength	Valid	7	0x07
bDescriptorType	ENDPOINT	5	0x05
bEndpointAddress	4 OUT [1]	4	0x04
bmAttributes.TransferType	Bulk	2	0x02
wMaxPacketSize	64	64	0x0040
bInterval		0	0x00

1. Given endpoint address is an example and can be changed without notice.

4.2.2.5 Interface 3 (VCOM2 Port)

Table 37: MDM\Configuration Descriptor\Interface2

Name	Value	Dec	Hex
bLength	Valid	9	0x09
bDescriptorType	INTERFACE	4	0x04
bInterfaceNumber	3	3	0x03
bAlternateSetting	0	0	0x00
bNumEndpoints	2	2	0x02
bInterfaceClass	Vendor-specific	255	0xFF
bInterfaceSubClass	Vendor-specific	255	0xFF
bInterfaceProtocol	Vendor-specific	255	0xFF
iInterface	3	3	0x03

Endpoint Descriptor

Table 38: MDM\Configuration Descriptor\Interface2\Endpoint Descriptor

Name	Value	Dec	Hex
bFunctionLength	Valid	7	0x07
bDescriptorType	ENDPOINT	5	0x05
bEndpointAddress	5 IN [1]	133	0x85
bmAttributes.TransferType	Bulk	2	0x02
wMaxPacketSize	64	64	0x0040
bInterval		0	0x00

1. Given endpoint address is an example and can be changed without notice.

Endpoint Descriptor

Table 39: MDM\Configuration Descriptor\Interface2\Endpoint Descriptor

Name	Value	Dec	Hex
bFunctionLength	Valid	7	0x07
bDescriptorType	ENDPOINT	5	0x05
bEndpointAddress	5 OUT [1]	5	0x05
bmAttributes.TransferType	Bulk	2	0x02
wMaxPacketSize	64	64	0x0040
bInterval		0	0x00

1. Given endpoint address is an example and can be changed without notice.

4.2.3 String Descriptor

The “Mdm“ enumeration uses the same string descriptors as the “MdmNet“ variant. For details see Section 4.1.3.